Moving Secure Software Assurance into Higher Education: A Roadmap for Change

Linda Laird, Nancy Mead, Dan Shoemaker

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
02 JUN 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Moving Secure Software Assurance into Higher Education: A Roadmap for Change	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stevens Institute of Technology,Castle Point on Hudson,Hoboken,NJ,07030-5991**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**31**	

3 Related Initiatives

1. Master of Software Assurance Reference Curriculum

II. Implementing a Practical Software Assurance Curriculum

III. Formulating and Disseminating Software Assurance Knowledge into Education

To Begin with – The Big Problem:

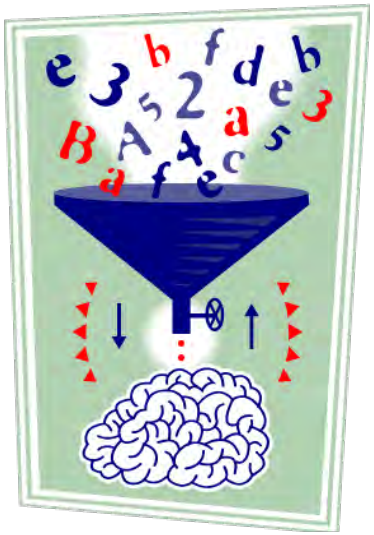
All Significant Systems Contain Defects

Defect Data By Application Domain – Reifer, 2004

Application Domain	Number of Projects	Error Range (Errors/ KESLOC)	Normative Error Rate (Errors/ KESLOC)	Notes
Automation	55	2 to 8	5	Factory automation
Banking	30	3 to 10	6	Loan processing, ATM
Command & Control	45	0.5 to 5	1	Command centers
Data Processing	35	2 to 14	8	DB-intensive systems
Environment/ Tools	75	5 to 12	8	CASE, compilers, etc.
Military -All	125	0.2 to 3	< 1.0	See subcategories
Airborne	40	0.2 to 1.3	0.5	Embedded sensors
Ground	52	0.5 to 4	0.8	Combat center
Missile	15	0.3 to 1.5	0.5	GNC system
Space	18	0.2 to 0.8	0.4	Attitude control system
Scientific	35	0.9 to 5	2	Seismic processing
Telecom	50	3 to 12	6	Digital switches
Test	35	3 to 15	7	Test equipment, devices
Trainers/ Simulations	25	2 to 11	6	Virtual reality simulator
Web Business	65	4 to 18	11	Client/server sites
Other	25	2 to 15	7	All others

So why don't we just get rid of all the defects?

- Why not just build everything to be highly reliable, safe, and secure? Why not make every system a "Trustable System?"



Out of 100 web app development projects .....

Summarized: The Issue:

- Software defects are currently a fact of life
- Software defects are avenues of security vulnerabilities that cyber criminals, terrorists, or hostile nations can exploit.
- We (THE ENTIRE INDUSTY) need to change the way we build systems
  - Decrease the number of defects
  - Tolerate faults and failures better
- HOW? Software Assurance addresses this problem
  - One HUGE part of the solution is formal education programs
  - These might start as low as middle school and flow upward all the way to advanced graduate study

So what is software assurance?

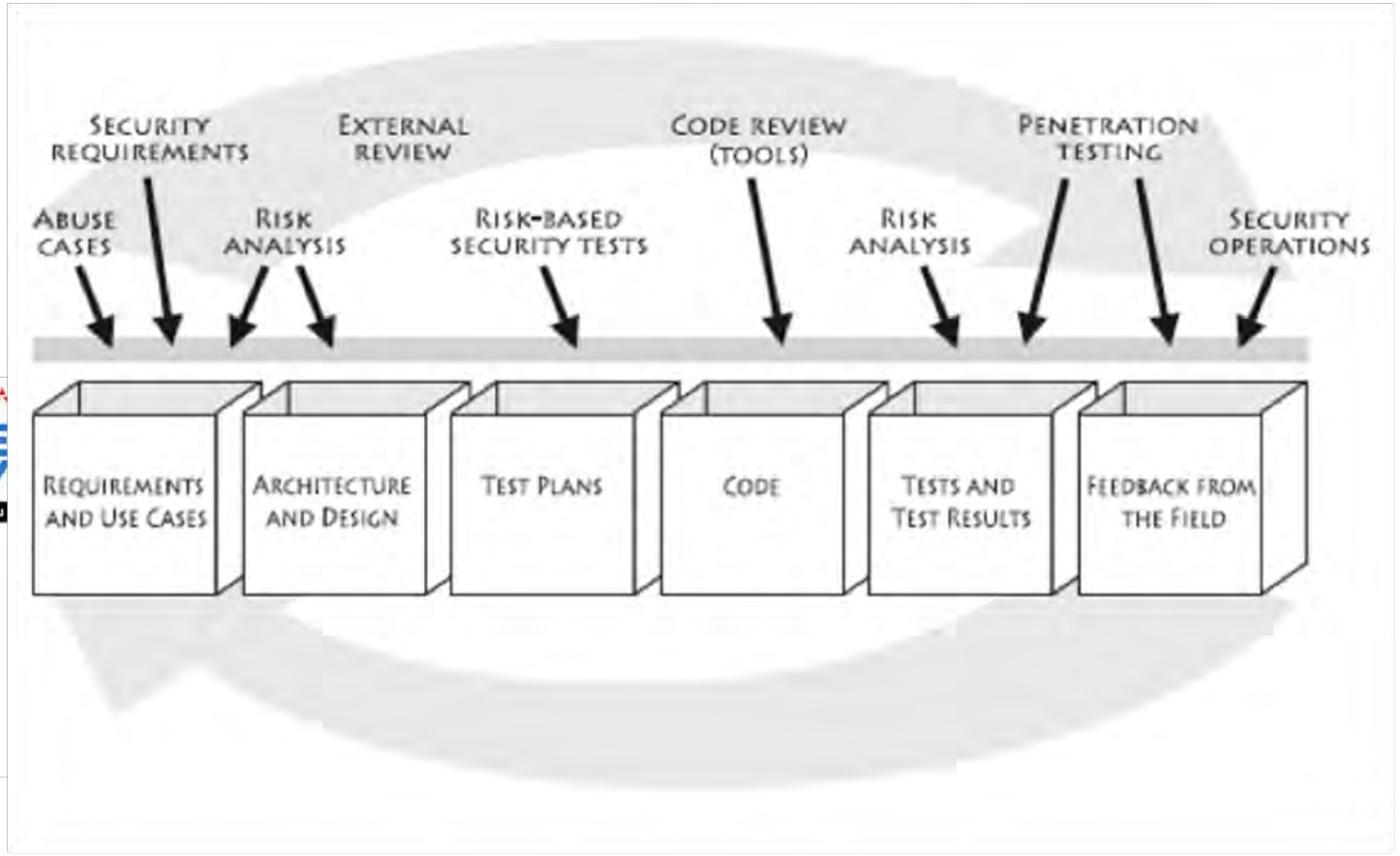
- *"Application of technologies and processes to achieve a required level of confidence that software systems and services function in the intended manner, are free from accidental or intentional vulnerabilities, provide security capabilities appropriate to the threat environment, and recover from intrusions and failures. "*
  - *Master of Software Assurance Reference Curriculum*

More Context: Software Assurance

- The OWASP Software Assurance Maturity Model (SAMM 1.0)
- 4 Business Functions, 3 Security Practices are defined
- The Security Practices cover all areas relevant to software security assurance

More Context: Touchpoints

- Gary McGraw's and Cigital's model



Three Problems with Education

- Essential SwA knowledge is cross cutting – as illustrated in the previous two charts
  - Generally agreed – the knowledge comes many fields such as software engineering, systems engineering, law, information assurance, security, ….
- It is not clear how to best deliver that knowledge to all of the relevant constituencies.
  - Educational institutions are very diverse
  - Computer education programs are also very diverse and focused at all levels from Community Colleges to PhD programs
- Few educators in our current classrooms have any more knowledge about the topic than the students they teach.
  - Most senior faculty got their degrees in the 1970s and 1980s
  - Very few PhDs have been produced
  - Teachers need 42 hours of things to talk about to offer a new course
  - Instructional materials are just coming out on the topic

The Last Problem with Education

- SwA – did not have an accrediting body or national society to underwrite its validity
- Programs of study are validated by adherence to commonly accepted models for the discipline
- That is – you cannot legitimately call yourself a program of study if your curriculum does not comply with the recommendations of:
  - Computer Science (ACM) – CS 2001/ CS 2008
  - Software Engineering (IEEE) – SE 2004 /MSWE 2009 (IEEE/ACM)
  - Information Systems (AIS) – IS 2002/MSIS 2006

From the Top – Initiative One: The Master of Software Assurance - MSwA

- Development of a master of software assurance reference curriculum (MSwA)
  - Lead by the Software Engineering Institute,
  - Supported by DHS's National Cyber Security Division,
  - Team members from 6 different academic institutions, both domestic and international
  - Reviewed by Industry, Government, and Academia

- Results:
  - Identifies the topics and the knowledge required to be an effective software assurance professional
  - Structures that set of topics into a comprehensive curriculum.
  - It has been approved by IEEE and ACM, and is available at http://www.cert.org/mswa/

Curriculum Contents: Key Knowledge Areas for Well-Educated Practitioner

- **Assurance Across Life Cycles** –life-cycle processes and development models for new or evolutionary system development, and for system or service acquisition.
- **Risk Management** - risk analysis and tradeoff assessment, and to prioritization of security measures.
- **Assurance Assessment** - analyze and validate the effectiveness of assurance operations and create auditable evidence of security measures.
- **Assurance Management** - make a business case for software assurance, lead assurance efforts, understand standards, comply with regulations, plan for business continuity
- **System Security Assurance** - incorporate effective security technologies and methods into new and existing systems.
- **System Functionality Assurance** - verify new and existing software system functionality for conformance to requirements and to help reveal malicious content.
- **System Operational Assurance** - monitor and assess system operational security and respond to new threats.

Initiative Two: Implementing the MSwA

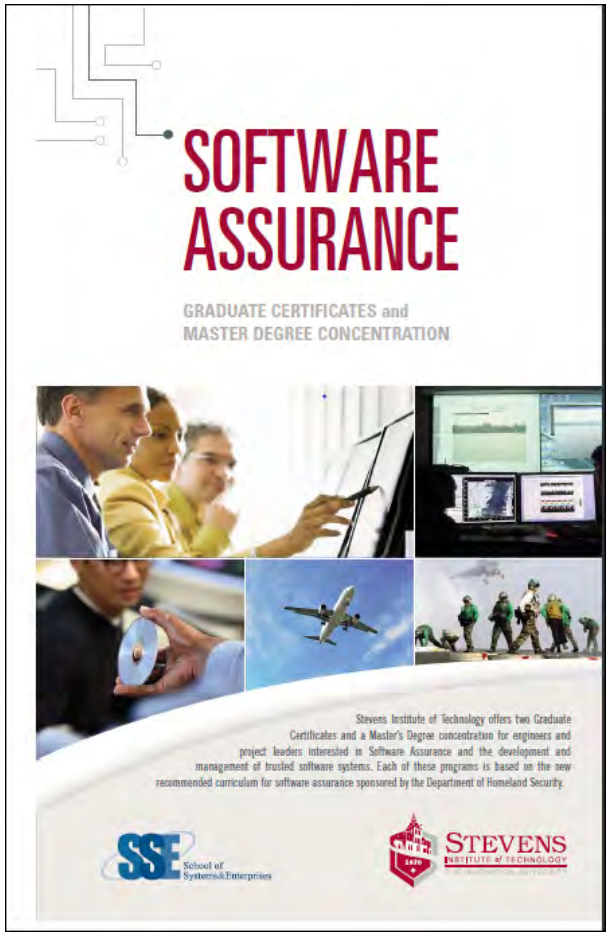
- Establishment of a new degree program is a very ambitious undertaking.

- Expectation that that some universities would elect to establish tracks or specializations in software assurance within existing master's degree programs rather than establishing a separate new degree program.

- Stevens Institute of Technology Software Assurance Program – proof of concept

Stevens Software Assurance Program

- 2 Graduate Certificates in Software Assurance
  - Development of Trusted Software Systems
  - Acquisition and Management of Trusted Software Systems
- Master's Degree in Software Engineering with a Concentration is Software Assurance
  - 10 required courses



Stevens' Implementation

- Advantages:
  - Three relevant programs:
    - Software Engineering (strong in traditional software engineering)
    - Computer Science (strong in traditional security)
    - Systems Security Engineering (strong in security from the systems perspective)
  - A Stevens faculty member was a member of the curriculum team
  - Motivated Software Engineering Faculty
    - The faculty believed every Steven's software engineering student should know how to engineer and build trustworthy (safe, secure, resilient, and reliable) systems.
  - Flexible Program Architecture
- Strategy:
  - integrate the software assurance curriculum into the existing software engineering curriculum, to the maximum extent possible.

Stevens' Issues

- Knowledge: Majority of the SWE faculty not particularly strong in security → Lots of individual learning and effort
- Effort: Significant amount of material needed to be developed and other material removed to make room.
  - 90% of work done in addition to normal workload
- No simple mapping from recommendations to curriculum:
  - Step by step approach through curriculum
- Overlaps between Software Assurance Curriculum and Systems Security Engineering and Computer Science
  - For SSE, additional material was added to support the curriculum, and these became part of the software assurance tracks as well.
  - For CS, there were three overlapping security courses, but the curriculum had room only for one. Selected material from the three was collapsed and additional material was added to create a new course.

Examples of Course Changes

- SSW 689: Software Safety and Reliability Engineering → SSW 689: Engineering of Trusted Software Systems
  - Added and Extended
    - Overarching model of trusted systems: secure, dependable, safe, and resilient
    - Trust Cases, Assurance Maturity Models
    - Threat Modeling
    - Misuse and Abuse Cases
    - Risk Management Frameworks
    - Trusted (and Secure) Architecture Patterns and Analysis
  - Decreased
    - Variety and detail of reliability models
    - Advanced topics in reliability testing

STEVENS Institute of Technology

Software Engineering

at Stevens Institute of Technology

SSE School of Systems&Enterprises

Doctoral Degree in Systems Engineering (60 credits, post Master's; minimum 30 research credits)

Master of Science in Software Engineering (SSW) (10 courses/30 credits)

Core Course Requirements

All students must take:

- SSW 540: Fundamentals of Quantitative Software Engineering
- SSW 533: Software Estimation and Measurement
- SSW 800: Masters Project

Additional required courses:

- SSW 564 Software Requirements Analysis and Engineering
- SSW 565 Software Architecture and Component-Based Design
- SSW 567 Software Testing, Quality Assurance and Maintenance
- 4 Electives (Advisor Approved)

SOFTWARE ENGINEERING

- SSW 540: Fundamentals of Quantitative Software Engineering
- SSW 533: Software Estimation and Measurement

Plus two of the following courses:

- CS 573 Fundamentals of CyberSecurity
- SSW 564 Software Requirements Analysis and Engineering
- SSW 565 Software Architecture and Component-Based Design
- SSW 567 Software Testing, Quality Assurance & Maintenance
- SSW 687 Engineering of Large Software Systems
- SSW 689 Software Reliability and Safety Engineering

SYSTEMS-CENTRIC SOFTWARE ENGINEERING

- SSW 540 Fundamentals of Quantitative Software Engineering
- SYS 625 Fundamentals of Systems Engineering
- SYS 612/MGT 609 Project Mgt. for Complex Systems
- SSW 565 Software Architecture and Component-Based Design

SOFTWARE DESIGN & DEVELOPMENT

- SSW 565 Software Architecture and Component-Based Design
- SSW 555 Agile Methods for Software Development
- CS 574 Object-oriented Design and Analysis
- CS 546 Web Programming
- *or* CS 548 Engineering of Enterprise Software Systems

SOFTWARE PROGRAM MANAGEMENT

- SSW 540 Fundamentals of Quantitative Software Engineering
- SSW 533 Software Estimation & Measurement
- SYS 612/MGT 609 Project Management for Complex Systems
- SSW 687 Engineering of Large Software Systems

DEPENDABLE SYSTEMS

- SSW 540 Fundamentals of Quantitative Software Engineering
- SSW 565 Software Architecture and Component-Based Design
- SSW 689 Software Reliability & Safety Engineering
- CS 573 Fundamentals of Cybersecurity
- *or* SES 602 Secure Systems Foundations

SOFTWARE ACQUISITION AND INTEGRATION

- SSW 540 Fundamentals of Quantitative Software Engineering
- SSW 564 Software Requirements Analysis and Engineering
- SSW 687 Engineering of Large Software Systems
- SYS 605 Systems Integration

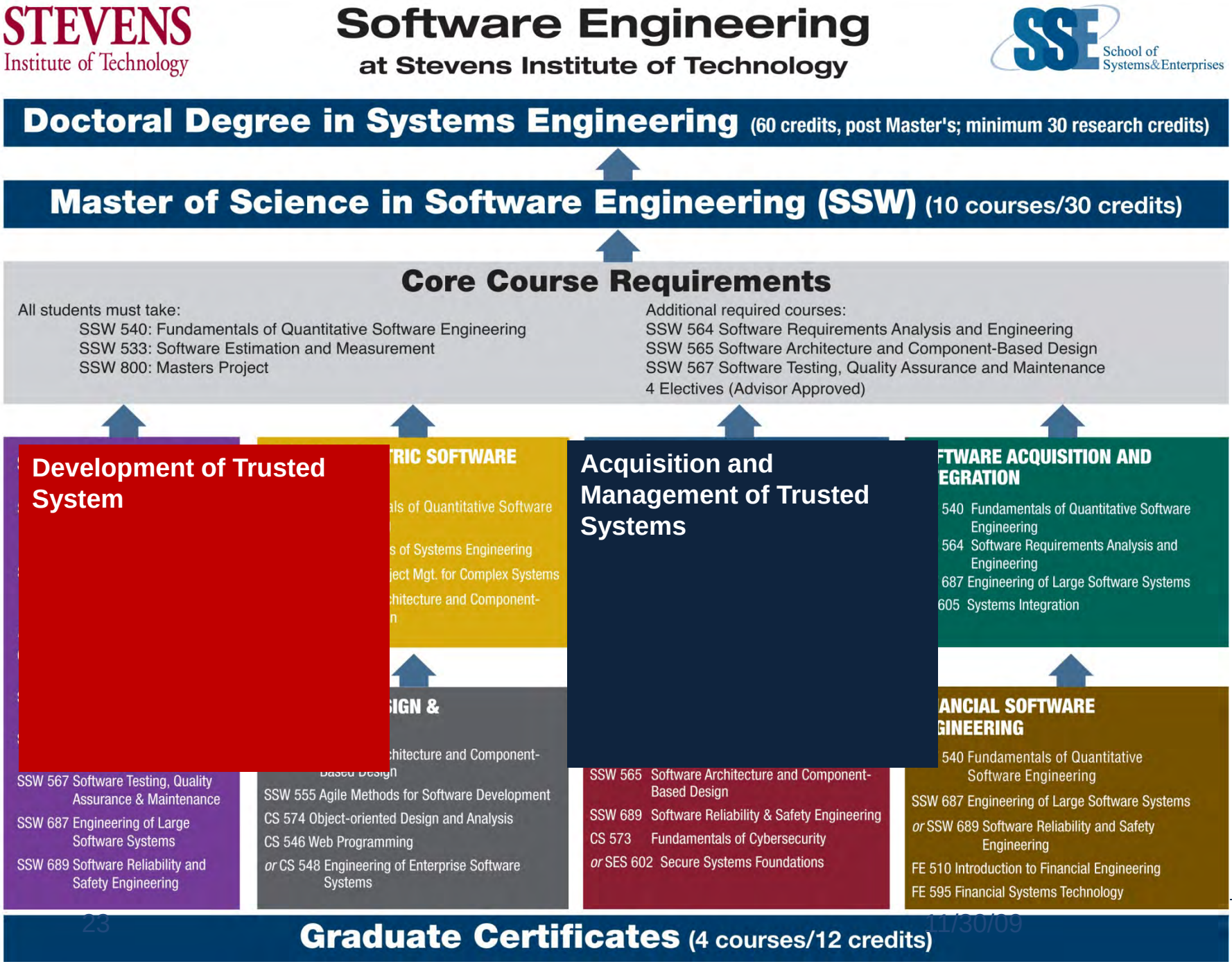
FINANCIAL SOFTWARE ENGINEERING

- SSW 540 Fundamentals of Quantitative Software Engineering
- SSW 687 Engineering of Large Software Systems
- *or* SSW 689 Software Reliability and Safety Engineering
- FE 510 Introduction to Financial Engineering
- FE 595 Financial Systems Technology

Graduate Certificates (4 courses/12 credits)

Results: Two Grad Certificates

- Development of Trusted Systems
  - SES 602: Secure Systems Foundations – Foundational security knowledge and technology from a systems perspective
  - SES 603: Secure Systems Laboratory – Hands-on lab that accelerates experience in systemic security issues
  - SSW 556: Software Development for Trusted Systems – How to develop systems without vulnerabilities and recognized vulnerabilities in existing software
  - SSW 689: Engineering of Trusted Software Systems: How to architect and design safe, reliable, secure, and resilient systems
- Acquisition and Management of Trusted Systems
  - SES 602: Secure Systems Foundations
  - SSW 533: Software Estimation and Measurement: How to estimate and measure the effort, reliability, and trustability of a system
  - SSW 564: Software Requirements Analysis and Engineering: How to elicit and write the right requirements
  - SSW 687: Acquisition and Management of Large Software Systems: How to acquire, integrate, and manage large scale developments


STEVENS
Institute of Technology
Software Engineering
at Stevens Institute of Technology
SSE School of Systems&Enterprises
Doctoral Degree in Systems Engineering (60 credits, post Master's; minimum 30 research credits)
Master of Science in Software Engineering (SSW) (10 courses/30 credits)
Core Course Requirements
All students must take:
SSW 540: Fundamentals of Quantitative Software Engineering
SSW 533: Software Estimation and Measurement
SSW 800: Masters Project
Additional required courses:
SSW 564 Software Requirements Analysis and Engineering
SSW 565 Software Architecture and Component-Based Design
SSW 567 Software Testing, Quality Assurance and Maintenance
4 Electives (Advisor Approved)
Development of Trusted System
Acquisition and Management of Trusted Systems
RIC SOFTWARE
ls of Quantitative Software
s of Systems Engineering
ject Mgt. for Complex Systems
hitecture and Component-
n
FTWARE ACQUISITION AND EGRATION
540 Fundamentals of Quantitative Software Engineering
564 Software Requirements Analysis and Engineering
687 Engineering of Large Software Systems
605 Systems Integration
SIGN &
hitecture and Component-
Based Design
SSW 555 Agile Methods for Software Development
CS 574 Object-oriented Design and Analysis
CS 546 Web Programming
or CS 548 Engineering of Enterprise Software Systems
ANCIAL SOFTWARE GINEERING
540 Fundamentals of Quantitative Software Engineering
SSW 687 Engineering of Large Software Systems
or SSW 689 Software Reliability and Safety Engineering
FE 510 Introduction to Financial Engineering
FE 595 Financial Systems Technology
SSW 567 Software Testing, Quality Assurance & Maintenance
SSW 687 Engineering of Large Software Systems
SSW 689 Software Reliability and Safety Engineering
SSW 565 Software Architecture and Component-Based Design
SSW 689 Software Reliability & Safety Engineering
CS 573 Fundamentals of Cybersecurity
or SES 602 Secure Systems Foundations
Graduate Certificates (4 courses/12 credits)

Result: Master's Degree and Stronger Program

- Master's Degree in Software Engineering with a Concentration in Software Assurance
  - Two Tracks:
    - Developing Trusted Systems – Developer Focused
    - Managing Trusted Systems – Acquisition and Management Focused

- Our Conclusion:
  - Stronger program. Hopefully, graduating more knowledgeable software engineers (with or without the software assurance tracks!)
  - See www.stevens.edu/software

Initiative Three: Supporting the Teaching Process

- Two- year project funded by the Department of Defense (DoD) and conducted at the University of Detroit Mercy **to identify, relate and catalogue** what is presently software assurance knowledge presently exists

- The **knowledge base** that was the product of this year long study
  - Documented and categorized all commonly accepted practices, principles, methodologies and tools for software assurance.
  - Incorporates as many lifecycle methodologies and tools for assuring software as could be identified.
  - This knowledge base is fully web accessible to anybody who wishes to use it

Initiative Three: Supporting the Teaching Process

- Nevertheless, the actual purpose this initiative was to ensure the teaching of secure software topics in all suitable education, training and awareness settings.
- In support of that goal, the project then packaged the contents of the knowledge base into discrete learning modules.
- These modules are meant to facilitate the efficient transfer of software assurance knowledge into all relevant teaching and learning settings.
  - They are appropriate for traditional graduate and undergraduate, community college and even high school education, as well as training and awareness applications. .

Standalone Teaching Modules

- Development of Secure Code
  - Risk Understanding
  - Threat Modeling
- Secure Sustainment of Code
  - Ethical hacking
  - Environmental monitoring and reporting
  - Risk analysis
  - Authorization
  - Change control
  - Patch management
- Acquisition of Secure Code
  - Acquisition initiation
  - secure specification
  - contract formulation and delivery management.

Initiative Three: Supporting the Teaching Process

- Each of the actual teaching modules incorporates a set of conventional learning support artifacts, which are easily recognizable to traditional educators.
- Every module includes
  - A table of learning specifications
  - Presentation slides for each concept contained in the module
  - A model evaluation process
  - Any relevant web-enabled supporting material
  - Videos
  - A model lesson plan
- All packaged onto an IPAD for easy portability
- See http://cybersecurity.udmercy.edu/

3 Related Initiatives

1. Master of Software Assurance Reference Curriculum

II. Implementing a Practical Software Assurance Curriculum

III. Formulating and Disseminating Software Assurance Knowledge into Education

Thank you. Questions?

Linda M. Laird –
linda.laird@stevens.edu

Industry Professor and Director of Software Engineering

School of Systems and Enterprises

Stevens Institute of Technology

Glossary

- DHS – Department of Homeland Security
- MSwA – Master of Software Assurance
- OWASP – Open Web Application Security Project
- SAMM – Software Assurance Maturity Model
- SES – Security Systems Engineering
- SwA – Software Assurance
- SSW – Software Engineering Program Designation at Stevens
- SWE – Software Engineering